**目 录**

[1实训介绍 1](#_Toc22823)

[2实训目的和要求 1](#_Toc2290)

[3实训设备及材料 1](#_Toc32268)

[4实训内容 1](#_Toc24383)

[4.1作品设计 1](#_Toc20567)

[4.2作品实现 1](#_Toc31853)

[4.3作品展示 1](#_Toc448)

[5实训总结 1](#_Toc32246)

# 1实训介绍

1.1项目背景

现即时聊天系统是目前网络上最为流行的通讯方式，而各种各样的即时通讯软件也层出不穷;服务提供商也提供了越来越丰富的通讯服务功能。随着互联网的发展，即时通讯的运用将日益广泛。

所开发软件名称:QQchat即时聊天系统

项目开发成员:李金航，林国龙，白杨，杨婷，王清元

# 2实训目的和要求

2.1实训目的

在完成“即时聊天系统”项目的可行性研究和需求分析的基础上，为了明确软件需求、安排项目规划与进度、组织软件开发与测试，项目小组经过深入讨论和分析之后，总结出该实训报告。

此实训报告对《即时聊天系统》软件的解决方案、功能分配、模块划分、程序的总体结构、输入输出和接口设计、运行设计等方面做了全面概括性说明，为该软件的详细设计奠定基础。

2.2实训要求

即时聊天软件开发（16学时）

开发一个带图形用户界面的类似QQ的即时聊天工具，要求具备以下功能：用户管理功能（注册、登录、头像、个人信息修改、查找用户）；聊天用户通讯录功能；用户私聊功能；多人聊天功能；聊天记录功能。

支持课程目标1、2、3、4、5。

思政元素：学习态度教育：认真对待实验，不弄虚作假，通过实验锻炼个人动手能力，加强对理论知识的理解。

2.3课程目标

**课程目标1：**能够通晓各种网络协议的封装格式、工作原理，熟悉网络编程的基本概念和方法，具备扎实的网络编程基础知识和专业知识。

**课程目标2：**能够对复杂网络工程问题进行分析，利用所学的数学、自然科学、专业知识和工程知识表述问题，分析问题，得到有效的结论。

**课程目标3：**能够根据需求设计网络系统，给出可行的解决方案，并能够在设计环节中体现创新意识，考虑社会、健康、安全、法律、文化以及环境等因素。

**课程目标4：**能够熟练使用一到两种开发工具，基于网络底层协议开发各种网络应用，具备较强的实践动手能力。

**课程目标5：**能够与他人合作进行复杂网络工程问题的研发，能够融入团队，具备团队合作精神。

# 3实训设备及材料

Visual Studio 2022

# 4框架设计

## **4.1****整体设计**

**4.1.1需求分析**

1.在Windows操作系统下实现服务器功能，在Windows 操作系统下实现即时聊天功能。

2.登录时只需要提供用户名即可，无需输入密码。

3.允许多人在线聊天 (至少能在六人之间进行相互聊天)4、聊天发送的消息包括: 用户名称、发送时间及正文。

4.某用户刚登录服务器时，服务器需对其发送实时在线用户列表。某用户登录或退出程序时都需要给服务器发送一个消息以通知其他用户。

5. 能够查看聊天记录。

**4.1.2路径设计**

**1.服务器运行**

|  |  |
| --- | --- |
| 主路径 | 可选路径 |
| 1.运行程序，显示登录窗口 |  |
| 2.输入用户账号 |  |
| 3.点击登录按钮，登录成功 | 3.登录失败  若登录的账号为空则进入注册界面 |
| 4.连接服务器 |  |
| 1. 连接服务器成功，提示用户登陆成功 | 5.连接服务器失败，提示用户连接失败，重新进入第二步 |
| 6.显示主窗口 |  |

**2. 客户端登录（本地）**

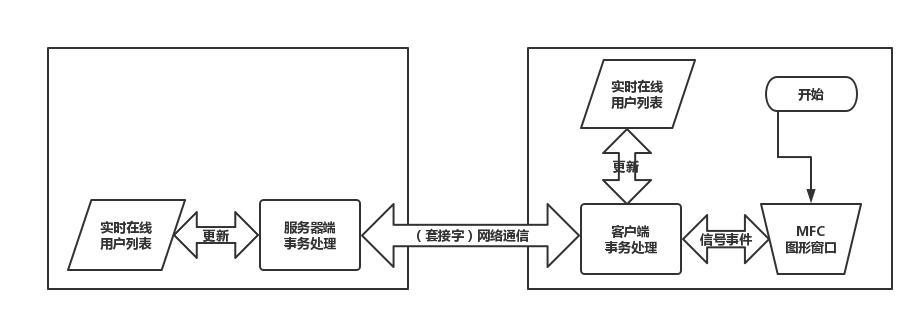
|  |  |
| --- | --- |
| 主路径 | 可选路径 |
| 1.运行程序，显示登录窗口 |  |
| 2.输入用户名 |  |
| 3.点击登录按钮，登录成功 | 3.登录失败  如果登录的用户名为空，重新进入第二步 |
| 4.连接服务器 |  |
| 5.连接服务器成功，提示用户连接成功 | 5.连接服务器失败，提示用户连接失败，重新进入第二步 |
| 6.显示主窗口 |  |

**3.客户端登录（远程）**

|  |  |
| --- | --- |
| 主路径 | 可选路径 |
| 1.侦听服务器发来的消息 |  |
| 2.解析消息 |  |
| 3.消息为远程主机登录  更新本地的在线用户列表  显示本地的在线用户 |  |

**4.客户端发送聊天信息**

|  |  |
| --- | --- |
| 主路径 | 可选路径 |
| 1.输入聊天信息 |  |
| 2.点击发送按钮或直接按回车键 | 2.没有输入消息则回到1 |
| 3.在用户界面上显示发出的消息 |  |
| 4.保存聊天信息至相应的聊天记录中 |  |

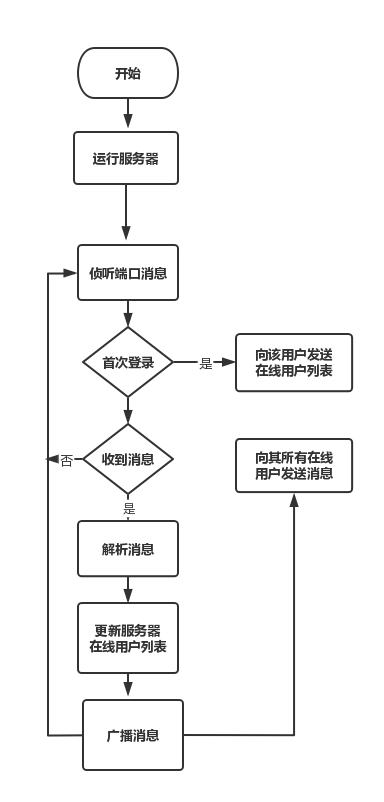
**4.1.3整体框架设计**

**服务器 客户端**

**4.2模块设计**

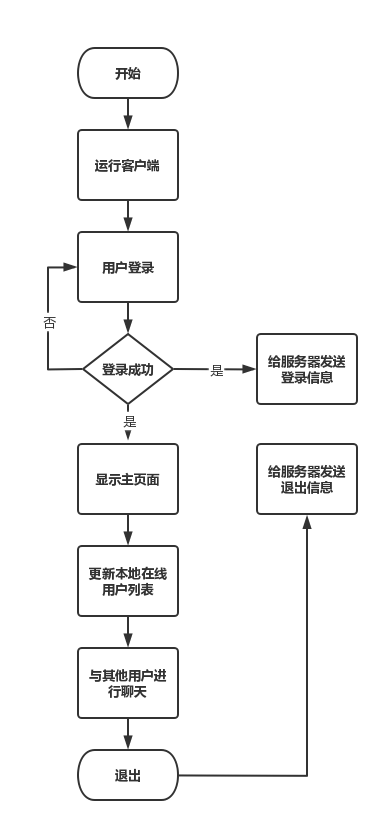
4.2.1服务器

服务器流程图如下：

z

4.2.2客户端

客户端流程图如下：



## 4.3作品实现

4.3.1

CgHeadDlg.cpp是一个c++实现文件(.cpp)在一个名为“OpenQQ”的应用程序中名为CCgHeadDlg的对话框类。代码定义对话框的行为和功能。

包括必要的头文件，如“stdafx.h”、“OpenQQ.h”、“cgheadlgh”和“afxdialogex.h”。

CCgHeadDlg类被声明和实现为cdialgex的子类。如图1-1所示



图1-1

重写DoDataExchange函数以启用对话框控件和变量之间的数据交换。如图1-2所示

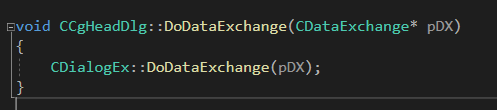


图1-2

使用BEGIN\_MESSAGE\_MAP和END\_MESSAGE\_MAP宏设置消息映射，将控制事件与其各自的事件处理程序函数关联起来。如图1-3所示

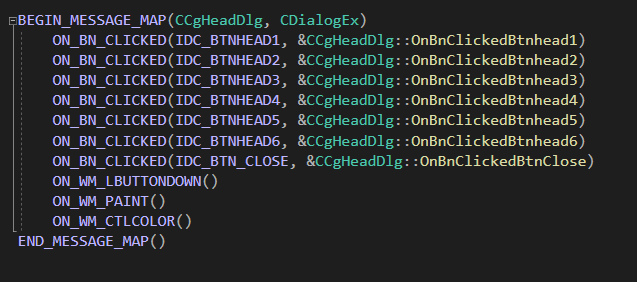


图1-3

OnInitDialog函数在初始化对话框时被调用。它设置对话框的初始状态，为按钮加载位图图像，并配置头部图标。如图1-4所示

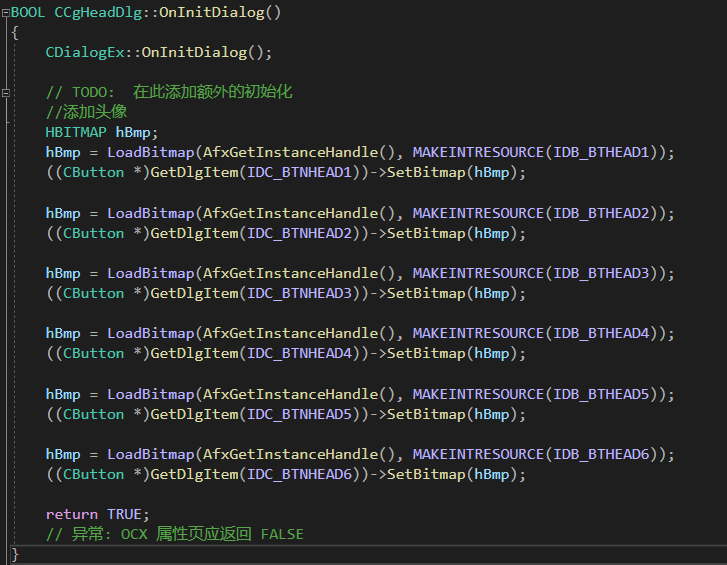


图1-4

事件处理函数，如onbnclickkedbtnhead1、onbnclickkedbtnhead2等，被实现来处理按钮点击和更新所选的头部图标。如图1-5、1-6所示

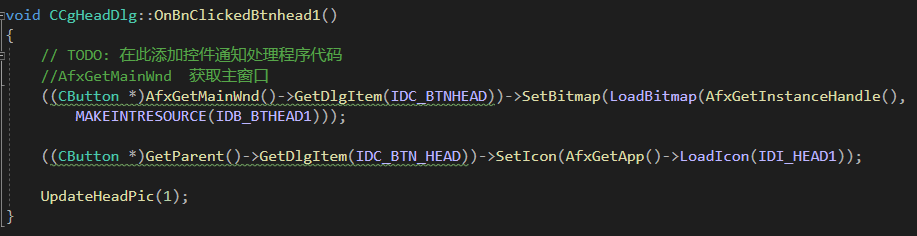


图1-5

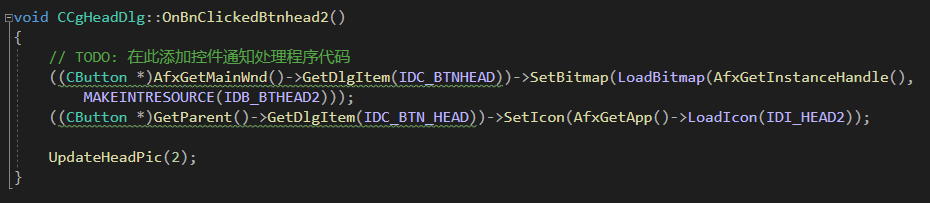


图1-6

其他函数，如OnCancel, UpdateHeadPic, onbnclickkedbtnclose, OnLButtonDown, OnPaint和OnCtlColor，处理对话框行为的不同方面，包括取消对话框，更新头部图片，关闭对话框，处理鼠标事件，绘制背景，控制对话框控件的颜色。如图1-7、1-8、1-9、2-1、2-2所示

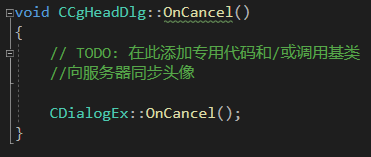


图1-7

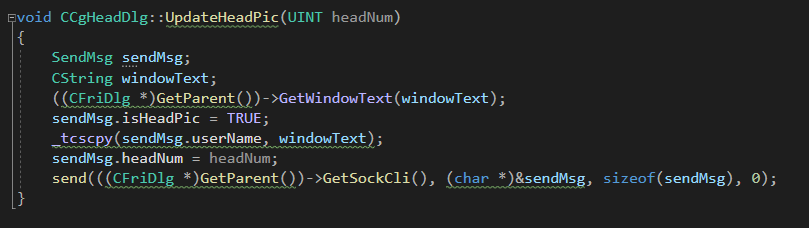


图1-8

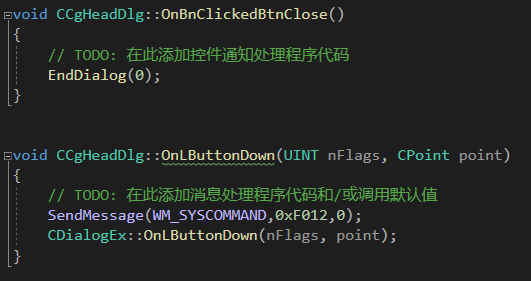


图1-9



图2-1

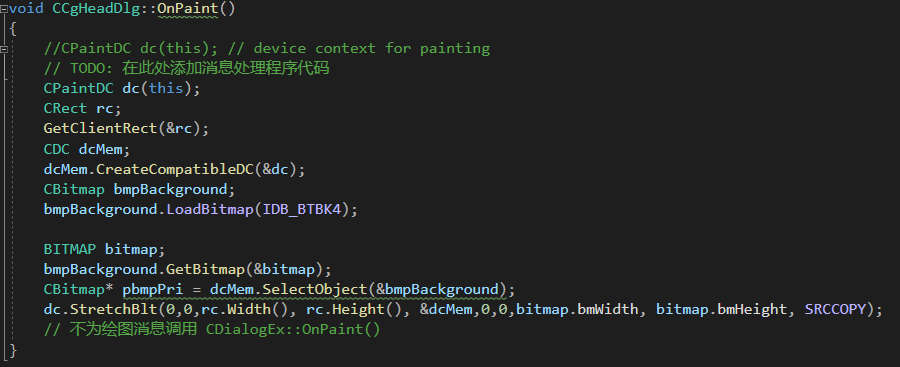


图2-2

ChatDlg.cpp是一个用于聊天窗口的对话框类的实现文件（ChatDlg.cpp）。这个对话框类用于显示聊天内容并发送消息。

该文件中的主要函数和功能包括：

OnBnClickedBtnsend()：当点击发送按钮时，获取编辑框中的文本，将消息发送到主窗口。如图2-3所示

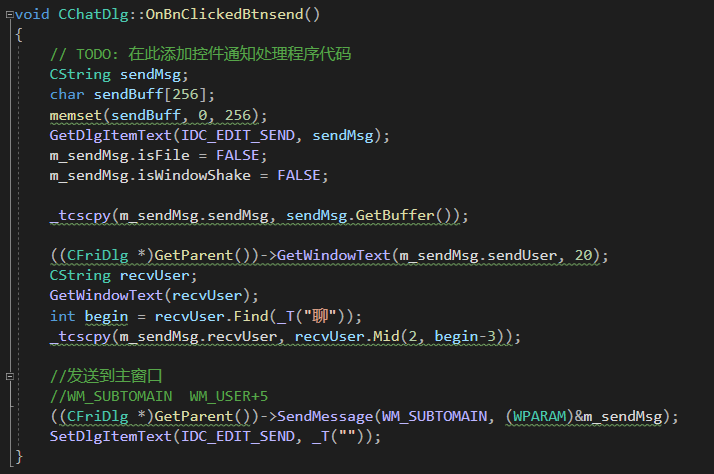


图2-3

OnSubWindow()：处理来自主窗口的消息，根据消息的类型设置相应的标志位。

如图2-4所示

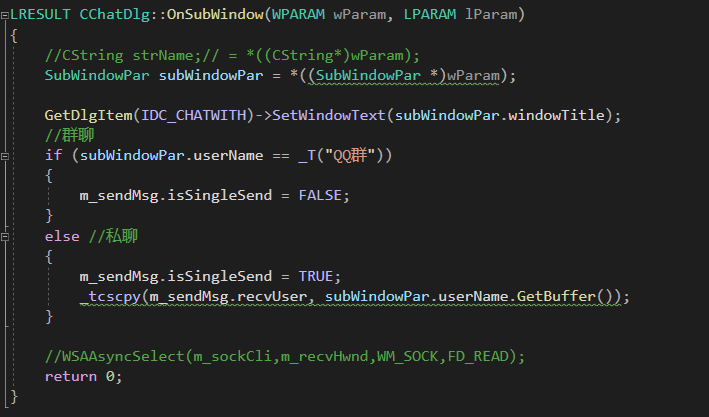


图2-4

OnInitDialog()：初始化对话框，加载按钮的位图资源。如图2-5所示

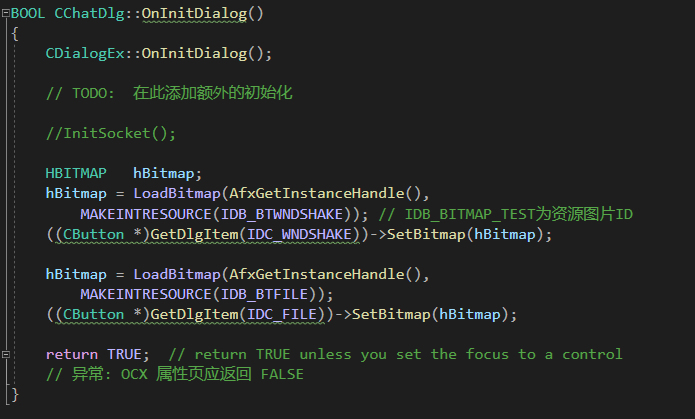


图2-5

OnCancel()：重写对话框关闭事件，销毁对话框并从父窗口的聊天对话框列表中移除该对话框。如图2-6所示

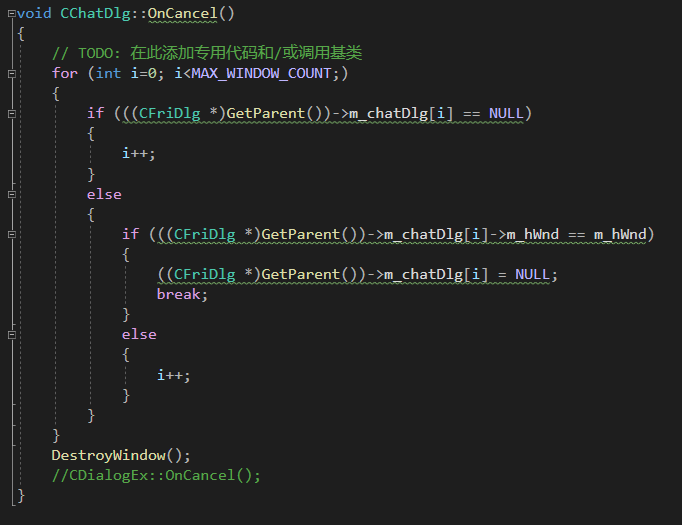


图2-6

OnLButtonDown()：处理鼠标左键按下事件，用于模拟窗口标题栏的拖动移动窗口。如图2-7所示

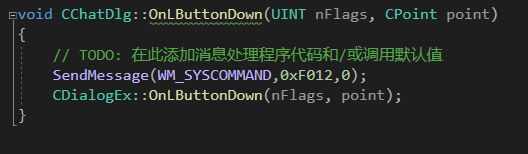


图2-7

OnBnClickedBtnClose()：点击关闭按钮时，关闭对话框。

如图2-8所示

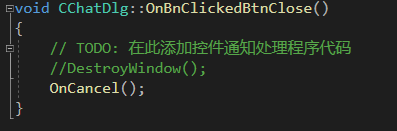


图2-8

OnPaint()：重绘对话框背景。如图2-9所示

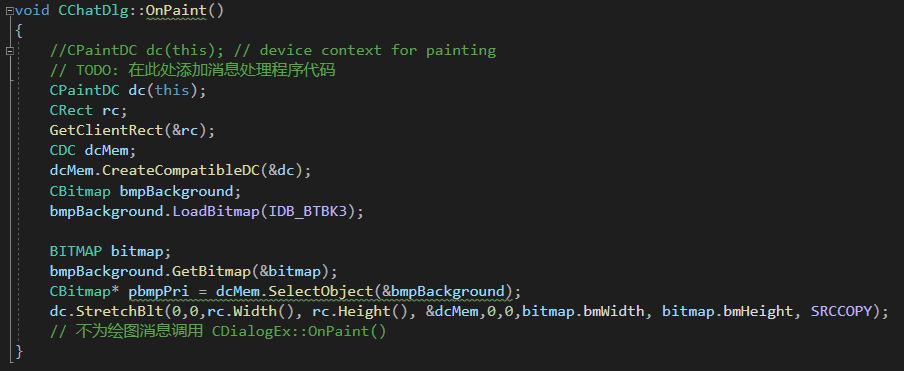


图2-9

OnBnClickedWndshake()：处理窗口抖动按钮的点击事件，发送窗口抖动消息到主窗口。如图3-1所示

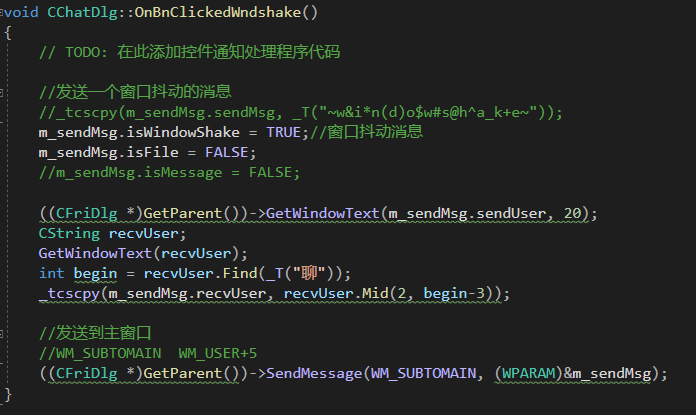


图3-1

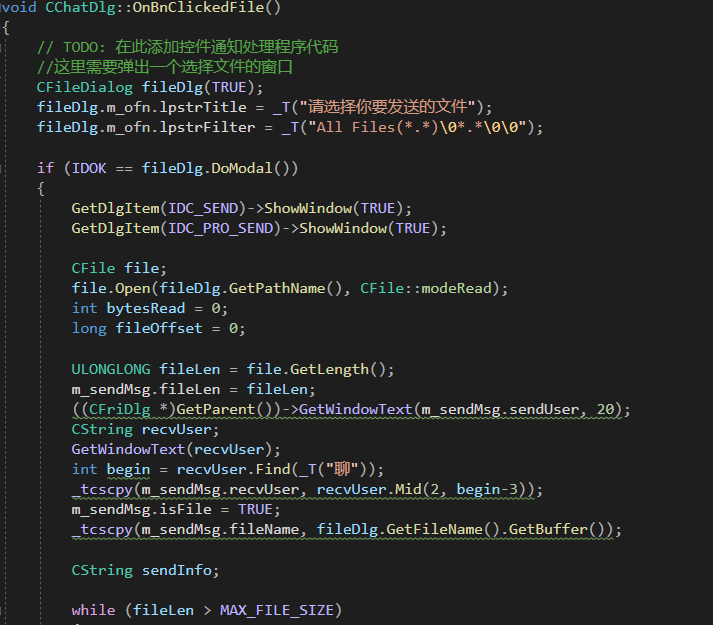
OnBnClickedFile()：处理发送文件按钮的点击事件，弹出文件选择对话框，选择文件后发送文件内容到主窗口。如图3-2所示

图3-2

请注意，这段代码只是对话框类的实现部分，缺少头文件和其他相关的代码，例如对话框类的声明、控件的创建和布局等。如果您需要完整的代码，请提供完整的代码文件或更多相关信息，以便能够更好地帮助您。

FriDlg.cpp是一个对话框类实现的朋友列表对话框。

下面是代码的概述:

代码包括必要的头文件，并定义了对话框类CFriDlg。如图3-3所示



图3-3

CFriDlg派生自cdialgex，它是MFC中对话框的基类。

代码实现了消息映射并在OnInitDialog函数中初始化了对话框控件。

OnInitDialog函数设置列表控件，为图标创建图像列表，并初始化其他控件和变量。如图3-4所示

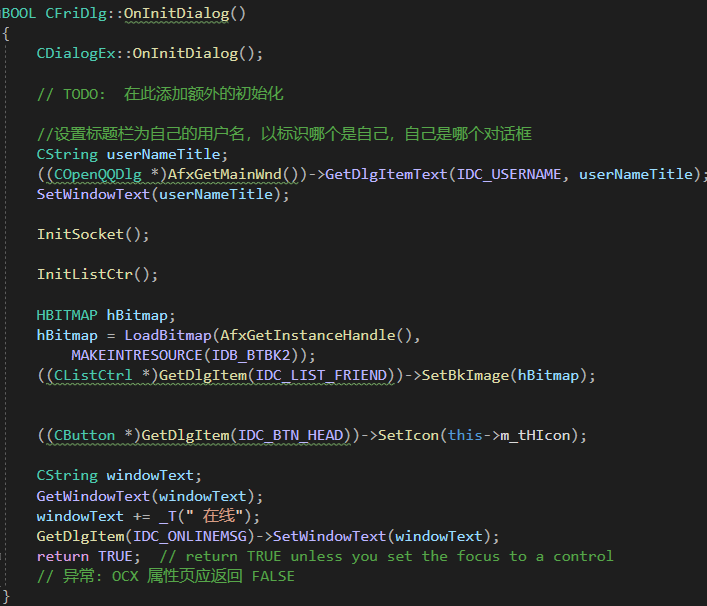


图3-4

InitSocket函数初始化与服务器通信的套接字。如图3-5所示

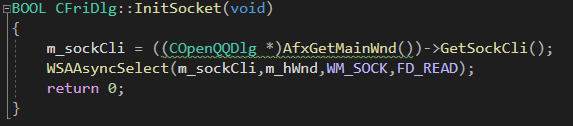


图3-5

OnSock函数处理来自服务器的传入消息，例如好友更新和聊天消息。它处理接收到的消息并相应地更新UI。如图3-6所示

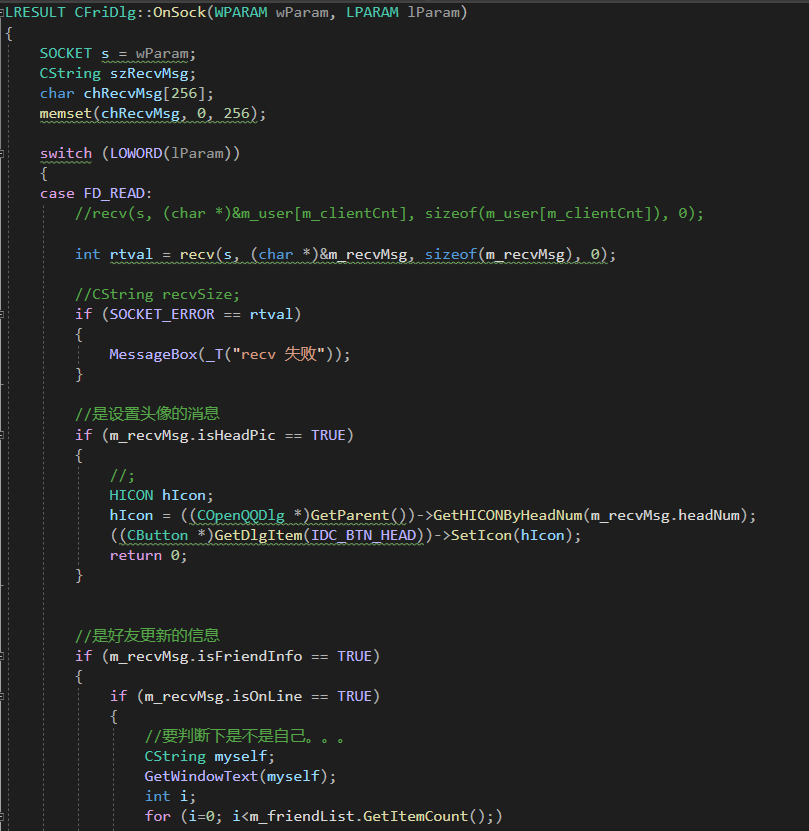


图3-6

OnNMClickListFriend函数处理在列表控件中单击好友时的事件。如果选择的朋友不存在，它会打开一个聊天窗口。如图3-7所示

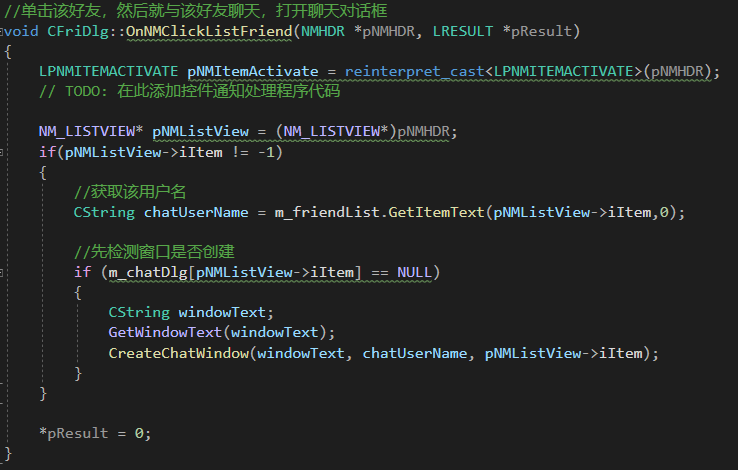


图3-7

其他函数处理UI事件，如最小化对话框到系统托盘和关闭对话框。

OnPaint和OnLButtonDown函数分别处理绘画和窗口行为。如图3-7、3-8所示

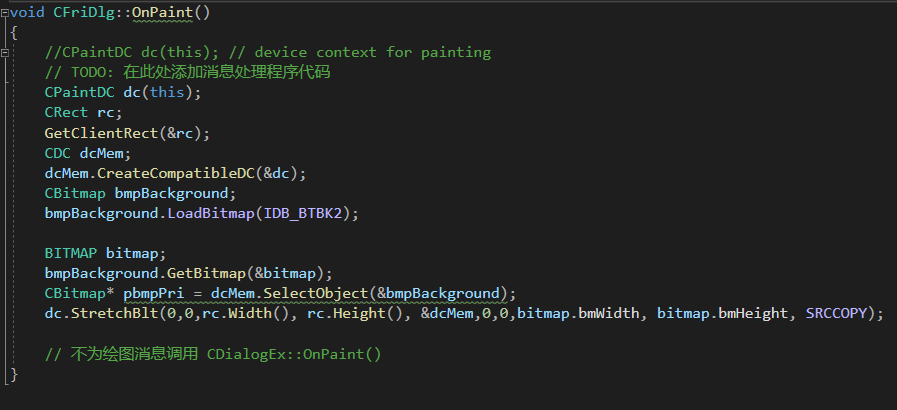


图3-7

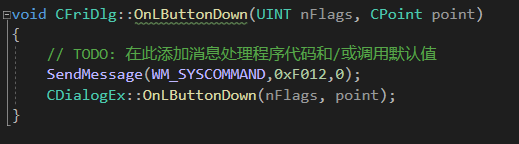


图3-8

注意:此代码片段可能无法单独编译或运行，因为它是较大应用程序的一部分，并且可能依赖于代码的其他部分。

OpenQQ.cpp是应用程序的主要代码文件（OpenQQ.cpp）。

该代码文件主要包含以下内容：

包含必要的头文件，包括"stdafx.h"（用于预编译的头文件）和"OpenQQ.h"（应用程序的主头文件）。

定义了一个名为COpenQQApp的应用程序类，该类派生自CWinApp类。

定义了COpenQQApp类的构造函数和成员函数InitInstance()。

实现了COpenQQApp类的消息映射宏（BEGIN\_MESSAGE\_MAP和END\_MESSAGE\_MAP），用于处理应用程序的消息和命令。如图4-1所示

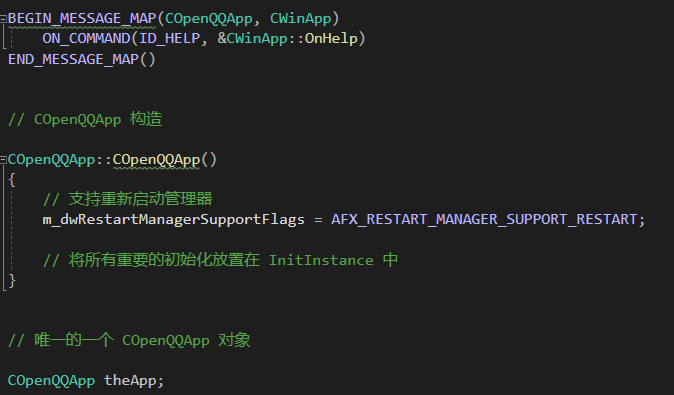


图4-1

在InitInstance()函数中进行了应用程序的初始化工作，包括：

调用WSAStartup()函数初始化Windows套接字库（Winsock）。

检查Winsock库的版本是否为2.2。

调用InitCommonControlsEx()函数初始化公共控件类。

创建COpenQQDlg对话框类的实例，并将其作为主窗口。

调用DoModal()函数显示对话框，处理对话框返回的结果。如图4-2所示



图4-2

在对话框关闭后，删除shell管理器的实例，并返回FALSE以退出应用程序。

总体上，这段代码是一个基于MFC的Windows应用程序的主要框架，它初始化应用程序并创建主对话框用于用户交互。

OpenQQDlg.cpp是应用程序主对话框的实现文件，名为“OpenQQDlg.cpp”。

代码包括与对话框的初始化、UI和功能相关的各种消息处理程序和函数。以下是主要组成部分的细分:

1.类定义:

CAboutDlg:这个类表示“关于”对话框。如图4-3所示

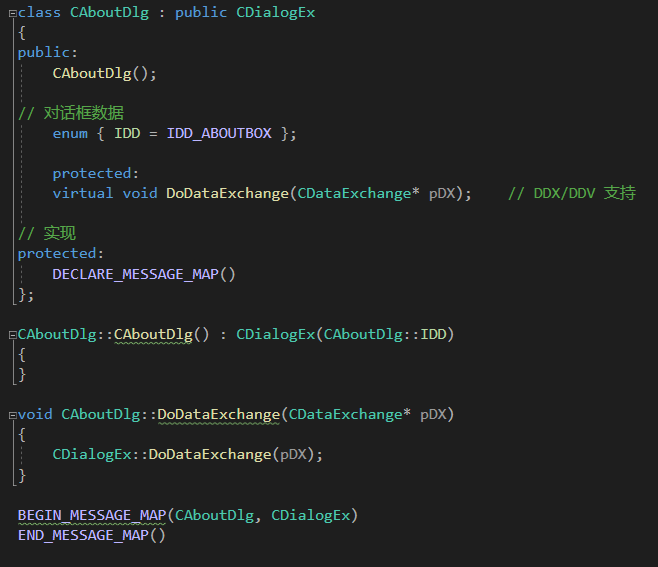


图4-3

COpenQQDlg: 这个类表示应用程序的主对话框。如图4-4所示

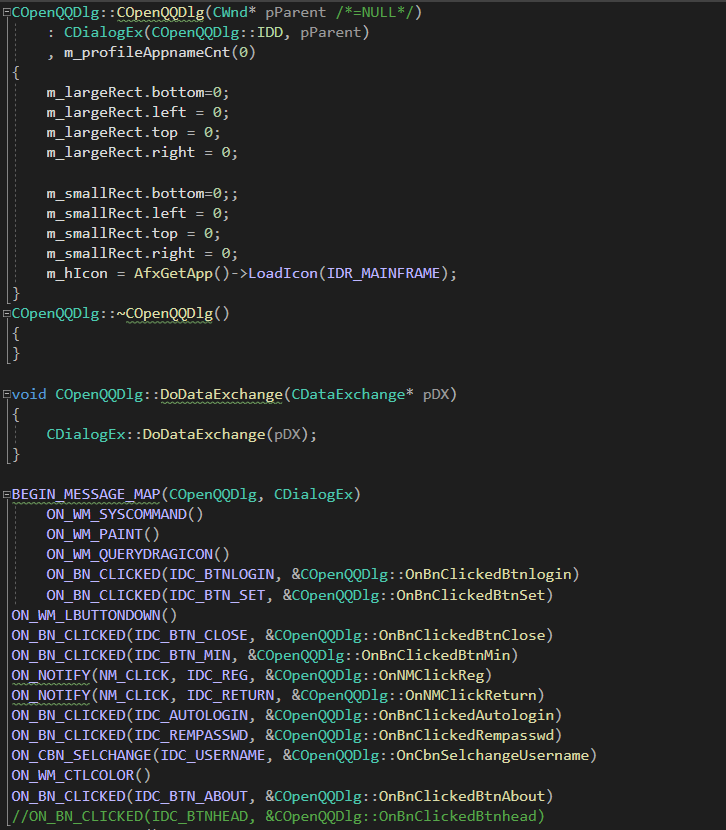
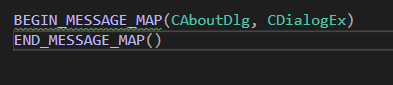


图4-4

2.消息映射:

BEGIN\_MESSAGE\_MAP宏用于将特定消息映射到对话框类中的相应函数。如图4-5所示

图4-5

初始化和UI:OnInitDialog:这个函数在第一次创建对话框时被调用，并处理各种组件的初始化，例如 设置图标、加载位图和配置控件。如图4-6所示

.

图4-6

4.事件处理程序:OnBnClickedBtnlogin:这个函数在点击“登录”按钮时被调用，并处理登录过程。如图4-7所示

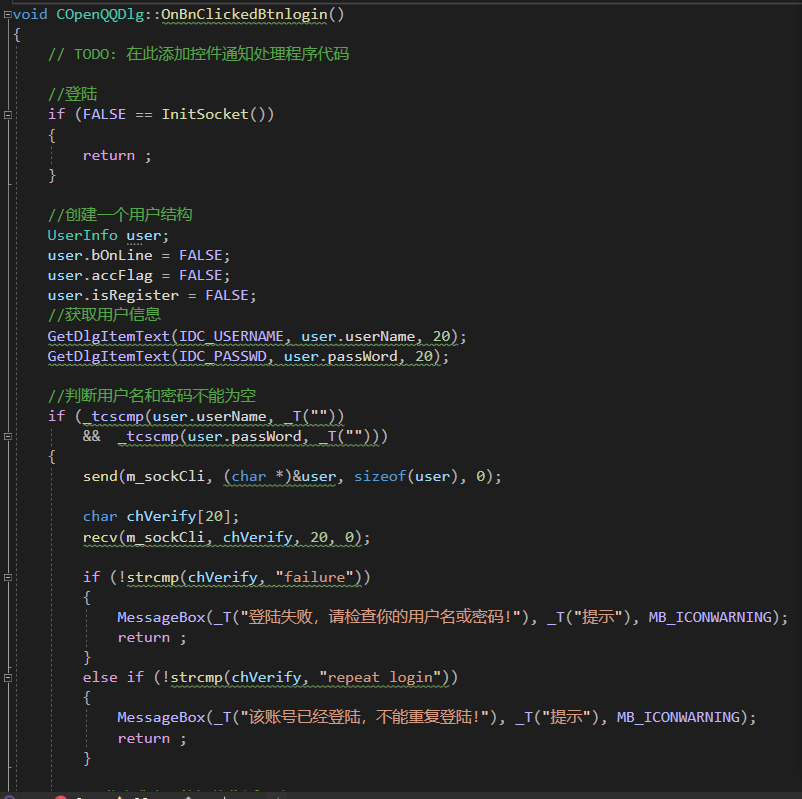


图4-7

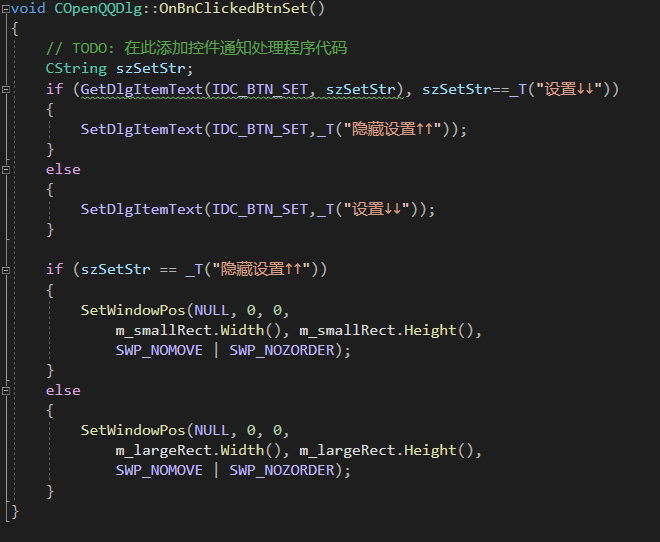
OnBnClickedBtnSet:这个函数在点击“↓↓”(设置)按钮并切换 对话框设置部分的可见性时被调用。如图4-8所示

图4-8

OnBnClickedBtnClose:这个函数在点击“关闭”按钮并关闭对话框时被调用。如图4-9所示

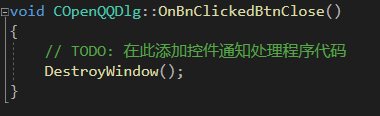


图4-9

OnBnClickedBtnMin:这个函数在点击“最小化”按钮并最小化对话框时被调用。如图5-1所示

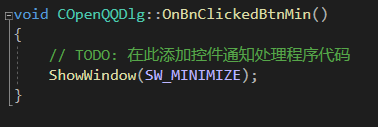


图5-1

OnNMClickReg:这个函数被调用，当点击(注册)链接并打开一个注册对 话框。如图5-2所示

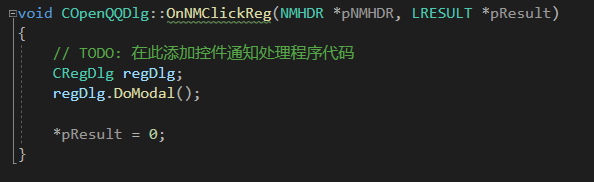


图5-2

OnNMClickReturn:当点击“返回”链接并显示密码恢复消息框时，该函数被调用。如图5-3所示

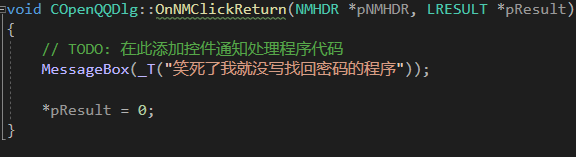


图5-3

onbnclickedautoologin:当单击“自动登录”复选框时调用该函数，并处理登录首选项的 存储。如图5-4所示

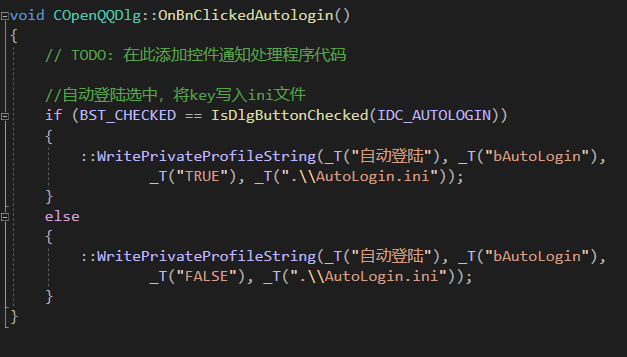


图5-4

5.套接字通信:

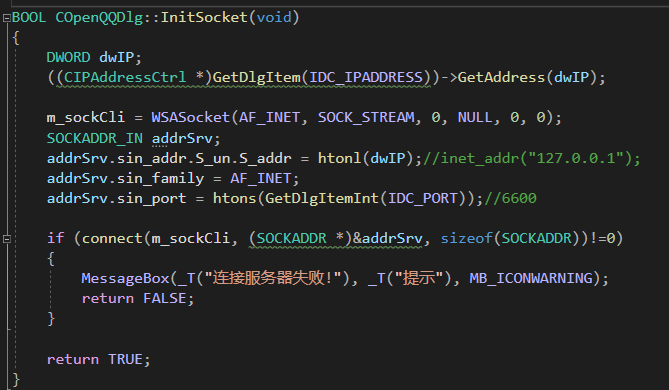
InitSocket:这个函数初始化与服务器通信的套接字。如图5-5所示

图5-5

GetSockCli:返回用于通信的套接字。如图5-6所示

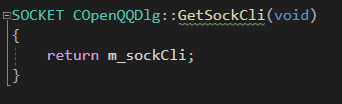


图5-6

总的来说，包括自动登录、注册和设置等功能。它还建立用于与服务器通信的套接字连接。

RegDlg.cpp是一个用于注册的对话框类 CRegDlg 的实现文件。该对话框提供了用户注册的功能。

代码中的 OnBnClickedReg 函数是注册按钮的点击事件处理函数，当用户点击注册按钮时，会执行该函数。函数首先通过父窗口的 InitSocket 函数初始化网络连接，然后创建一个 UserInfo 结构体对象，填充用户的注册信息。接下来，函数发送用户信息到服务器端进行注册验证，并接收服务器返回的注册结果。根据注册结果弹出相应的消息框提示用户注册成功或失败。如图5-7所示

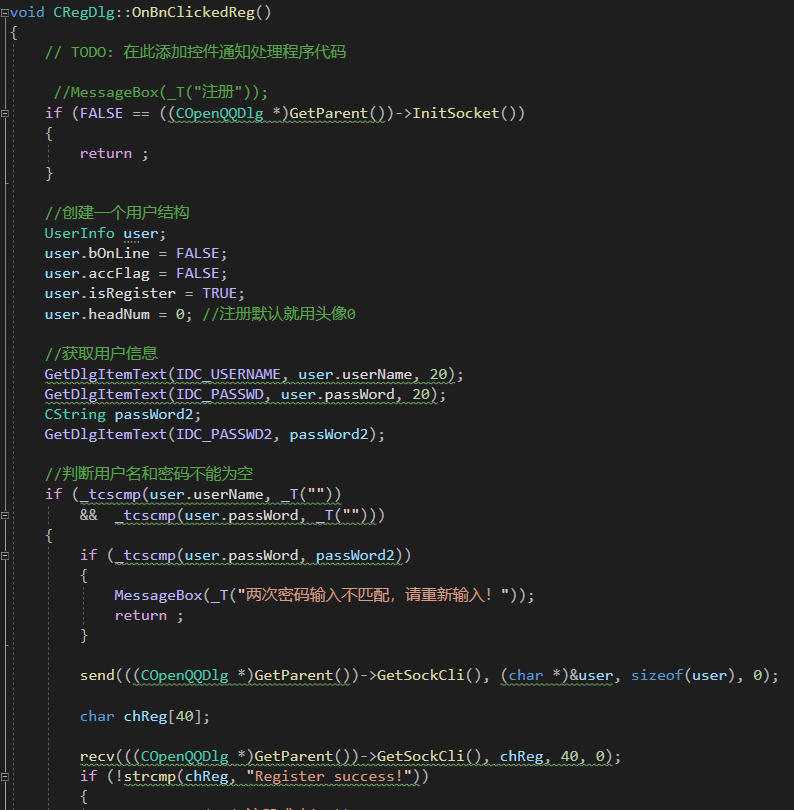


图5-7

OnBnClickedBtnClose 函数是关闭按钮的点击事件处理函数，当用户点击关闭按钮时，会执行该函数。函数调用 EndDialog 函数关闭对话框。如图5-8所示

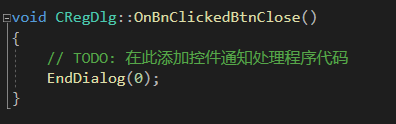


图5-8

OnLButtonDown 函数是鼠标左键按下事件处理函数，用于实现对话框的移动。当用户按下鼠标左键时，会发送系统消息 WM\_SYSCOMMAND，参数为 0xF012，以实现对话框的移动。如图5-9所示

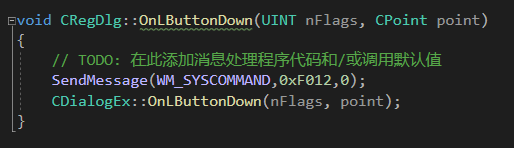


图5-9

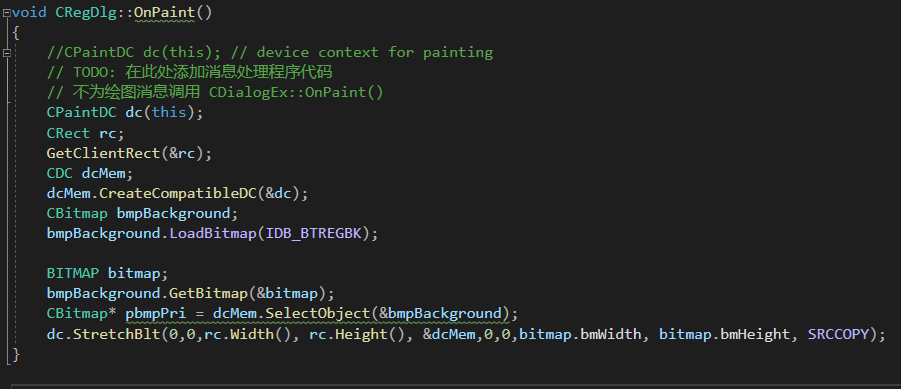
OnPaint 函数用于绘制对话框的背景图片。在函数中，首先创建一个兼容的设备上下文对象 dcMem，然后加载背景图片资源 IDB\_BTREGBK，并进行绘制。如图6-1所示

图6-1

OnCtlColor 函数用于设置对话框中控件的颜色。在函数中，判断控件类型是否为静态控件，如果是，则设置文字的前景色为黑色，并设置背景透明，返回透明画刷。如图6-2所示

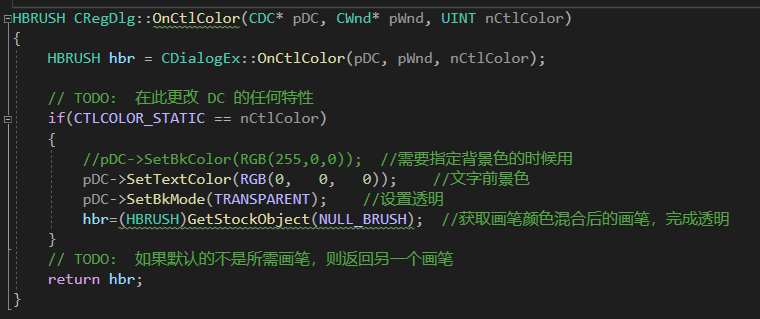


图6-2

这段代码基于 MFC（Microsoft Foundation Class）框架，用于构建 Windows 平台上的图形用户界面（GUI）应用程序。它包含了对话框类的定义和相应的消息处理函数，通过处理用户的操作来实现注册功能，并提供界面美化的效果。

## **4.4作品展示**

**4.4.1基础设置**

1.打开服务器，进行服务。如图6-3所示



图6-3

2.打开登录界面，功能有注册账号、找回密码、还有自动登录之类的设置。隐藏设置是要设置IP地址和端口用于通信。如图6-4所示



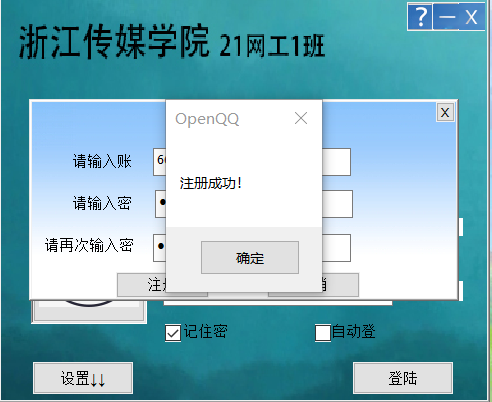
图6-4

3.点击注册账号，进行注册。如图6-5所示



图6-5

注册成功。如图6-6所示



**图6-6**

**4.4.2聊天软件的使用**

1.登录账号，界面出现。如图6-7所示



图6-7

1. 服务器端显示上线，服务器可以记录上线下线时间。如图6-8、6-9所示

图6-8

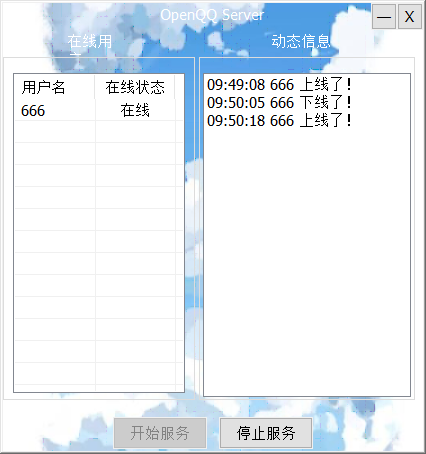


图6-9

3.点击头像可以选择更换头像。如图7-1所示



图7-1

**4.4.3 开始聊天**

1.1聊天界面如下。如图7-2所示

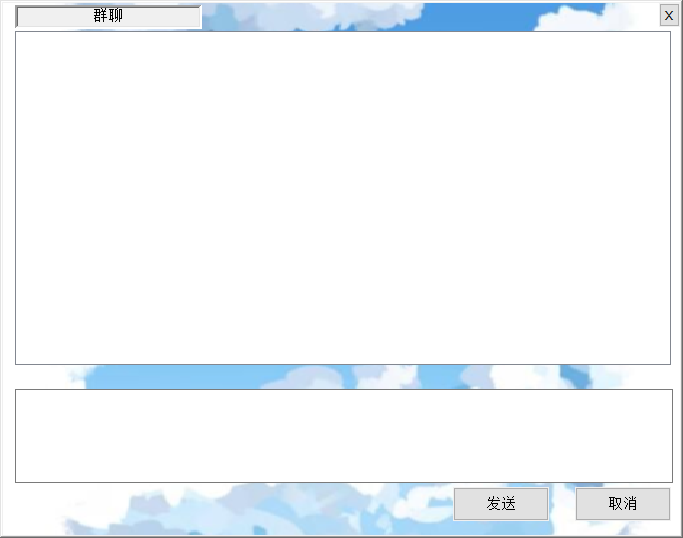


图7-2

开始发送信息。如图7-3所示

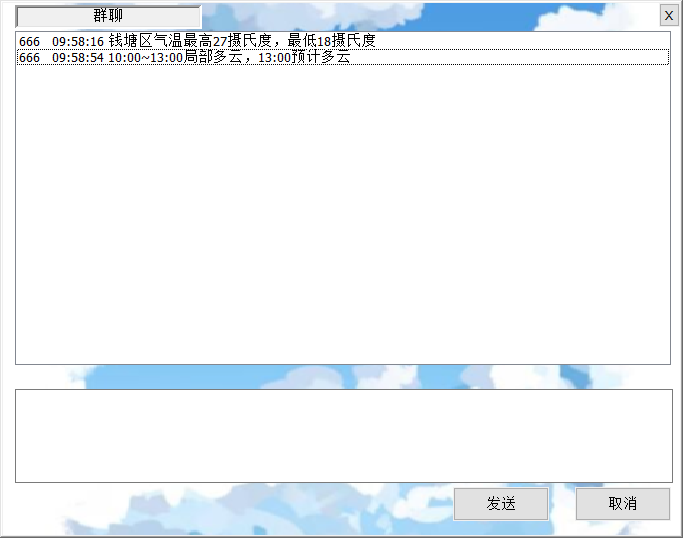
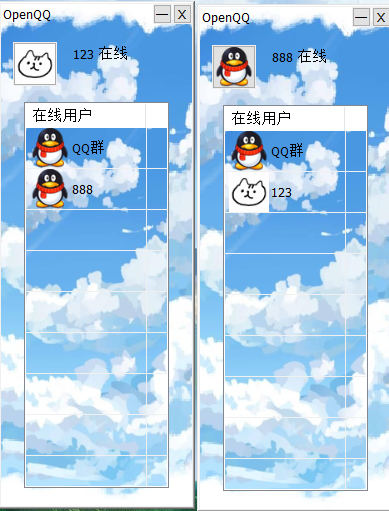


图7-3

**1.2多人聊天**

同时登陆两个账号进行聊天测试。如图7-4、7-5所示

 图7-4

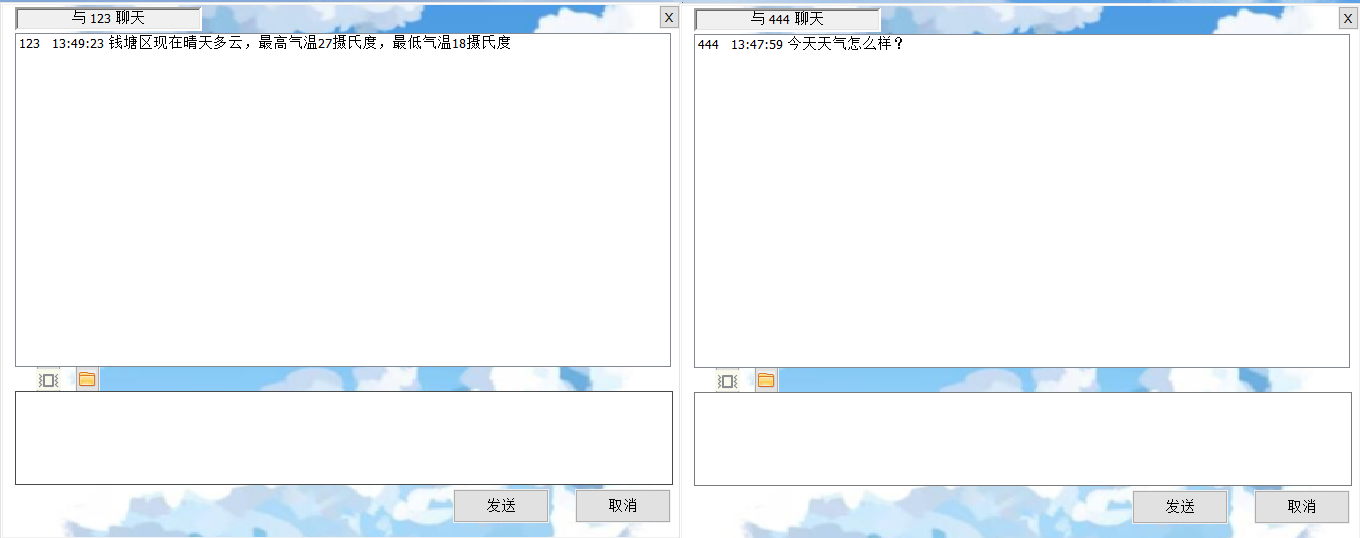


图7-5

# 5实训总结

我们完成了一个MFC的局域网聊天的网络编程项目，整个项目的完成过程中有许多值得总结和反思的地方。

首先，要注意项目的需求分析和功能设计。我们在进行项目开发的时候，在团队协作中需要明确每个人的责任和所需的技术能力，对于项目的需求和功能实现要进行仔细的分析和设计，以便于实现更好的协作。

其次，要注重项目架构和代码质量。在项目开发过程中，遵循良好的架构和代码规范能够增加代码的可读性和可维护性，减少代码的重复度和出错率，提高项目的整体质量。

再次，要反思项目的过程和结果，以便下一步做到更好。在项目的开发过程中，我们应该及时记录问题、收集反馈，总结经验教训，以便于后续的改进和升级。

最后，要团队协作，达成项目目标。一个好的项目需要团队协作，要注重团队内部的沟通和协作，减少误解和耽误，互相理解和支持，共同达成项目的目标。

MFC的局域网聊天的网络编程项目是一个很好的学习和实践机会，通过这个项目我们收获了很多宝贵的经验和教训，相信我们在今后的开发中会做得更加优秀。